EARTHQUAKE ENGINEERING AND STRUCTURAL DYNAMICS
Earthquake Engng. Struct. Dyn. **27**, 1225–1245 (1998)

NEURAL NETWORKS FOR STRUCTURAL CONTROL OF A BENCHMARK PROBLEM, ACTIVE TENDON SYSTEM

KHALDOON BANI-HANI[†] AND JAMSHID GHABOUSSI[*,‡]
Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

SUMMARY

Methodology for active structural control using neural networks has been proposed by Ghaboussi and his co-workers[1–8] in the past several years. The control algorithm in the mathematically formulated methods is replaced by a neural network controller (neuro-controller). Neuro-controllers have been developed and applied in *linear* and *nonlinear* structural control. Neuro-controllers are trained with the aid of the emulator neural networks. The emulator neural network is trained to learn the transfer function between the actuator signal and the sensor reading and it uses that past values of these quantities to predict the future values of the sensor readings. In this paper, we apply the previously developed neuro-control method in the benchmark problem of the active tendon system. The emulator neural network is developed and trained using the *evaluation* model given in the benchmark problem which is considered to be the true representation of the active tendon system. However, a *reduced-order* model has been developed and used, along with the emulator neural network, to train the neuro-controller. The *evaluation* model represents the three story steel frame structure, including the actuator dynamics. The absolute acceleration of the first floor and the actuator piston displacement are used as feedback. Three neuro-controllers, with different control criteria, have been developed and their performances have been evaluated with the prescribed performances indexes. The robustness of the neuro-controllers in the presence of some severe uncertainties, has also been evaluated. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: active control; neural networks; structures; earthquake engineering; dynamics

INTRODUCTION

Extensive research in the active control of civil engineering structures over the past few years have resulted in various algorithms, strategies and devices.[9] Most of these control methods require an accurate identification technique that can construct a precise mathematical model for the dynamics of the controlled system. Therefore, these methods can be called model-based control methods or mathematically formulated methods. On the other hand, control methods which acquire their capabilities through learning and adaptation, such as neuro-controllers and neuro-fuzzy-controllers, can be considered non-model-based methods or intelligent methods or adaptive methods. In spite of the remarkable developments in the field of structural control, no direct comparative study have been made between various proposed control methods. In this paper the neural network-based structural control method is evaluated, as part of a comparative study between the structural control methods initiated by the Structural Control Committee of ASCE.

Structural control methods, utilizing the learning capabilities of neural network have been developed by Ghaboussi and his co-workers. A neuro-control method based on the inverse transfer function was developed and applied in an experimental study of the actuator dynamics and delay compensation.[7] A neuro-control method which utilizes an emulator neural network (neuro-identifier) in its training, was

* Correspondence to: Jamshid Ghaboussi, Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.
† Research Assistant
‡ Professor of Civil Engineering

CCC 0098–8847/98/111225–21$17·50

Received 13 May 1997
Revised 2 December 1997

developed and applied in *linear* and *non-linear* structural control.[1–6] A similar method has been proposed by Chen *et al.*[10] Unlike the conventional control algorithms where the control task is explicitly formulated, in the neural network-based structural control methods the neuro-controller *learns* the control task. The neuro-controller acquires the knowledge of structural control from a set of training cases and stores that knowledge in its connection weights. One of the attractions of neural networks is that they are capable of learning complex non-linear relationships. It is for this reason that neural network-based control methods are equally effective in non-linear as well as in linear control problems.

In this study, an emulator neural network and three neuro-controllers, based on different control criteria, were developed and trained using the *evaluation* model described in the SIMU-LINK[11] program provided for the benchmark problem. The effectiveness of the trained neuro-controllers have been evaluated through two sets of criteria; the root mean squares (rms) of the responses of the structure when it is subjected to excitation of a stationary random process with a spectral density function defined by the Kanai–Tajimi spectrum; and, the peak responses when the structure is subjected to the compressed 1940 E1 Centro NS earthquake record and the compressed 1968 Hachinohe NS earthquake record. Finally, a study of the robustness of the neuro-controllers has been conducted and reported.

NEURO-CONTROL METHOD

Neuro-controller is a neural network which replaces the control algorithm in the mathematically formulated control methods. A typical neuro-controller is shown in Figure 1. Neuro-controllers can either be implemented in hardware or simulated in software, in the latter case, the neuro-controller is in the form of a software, residing in the control computer. Similar to the other control algorithms, the neuro-controller receives the feedback signal from the sensor (or sensors) at its input layer and issues an appropriate signal to the actuator from its output layer. In the software implementation of the neuro-controller, the sensor data is received at discrete-time intervals, referred to as the sampling periods, T_s. The output of the neuro-controller is also sent to the actuator at the same discrete sampling periods. In addition to the latest sensor reading, the input to the neuro-controller also consists of the history of both the sensor readings and the actuator ram displacement, at several previous sampling periods. In the present implementation of the neuro-controller, we have used two sensors and one actuator. The sensors consist of an accelerometer measuring the horizontal absolute acceleration at the first floor, $\ddot{x}_{a1}$, and a rigidly mounted LVDT measuring the actuator piston displacement, x_p.

The emulator neural network must be developed and trained before training of the neuro-controller. The emulator neural network learns the transfer function between the actuator signal (the signal going from the computer, where the control algorithm resides, to the actuator) and the output of the sensor measuring the response of the structure. The emulator neural network used in this study is shown in Figure 2. This transfer function includes the knowledge of the structural behaviour, as well as the knowledge of the actuator dynamics. In previous studies, Joghataie and Ghaboussi[5] and Ghaboussi and Joghataie[4] have developed and used a coupled model of the structure/actuator system. Similar coupled model of the structure/actuator system has been incorporated in preparing the *evaluation* model of the benchmark problem.[12] In addition to providing a path for training of neuro-controller, the emulator neural network also allows the forecasting of the response of the structure a short time period into the future, so that the control may be based on a time-averaged criterion.

In a digital control setting, the neuro-controller learns a relationship that generally can be described in the following manner. Consider a controlled structure subjected to a load vector p and actuator signal vector u and let y and z be the vector of sensor readings and output vector, respectively. The relationship that the neuro-controller learns must exist, otherwise the training process will not converge. We assume that the following function defining the future value of the actuator signal, subject to some constraint, does exist:

$$u_{k+1} = f_{nc}(y_k, y_{k-1}, \dots, y_{k-l}, u_k, u_{k-1}, \dots, u_{k-m}, P_k, P_{k-1}, \dots, P_{k-n}) \tag{1}$$
$$\text{subjected to } f_c(z_{k+1}, z_{k+2}, \dots, z_{k+p}) < \varepsilon$$

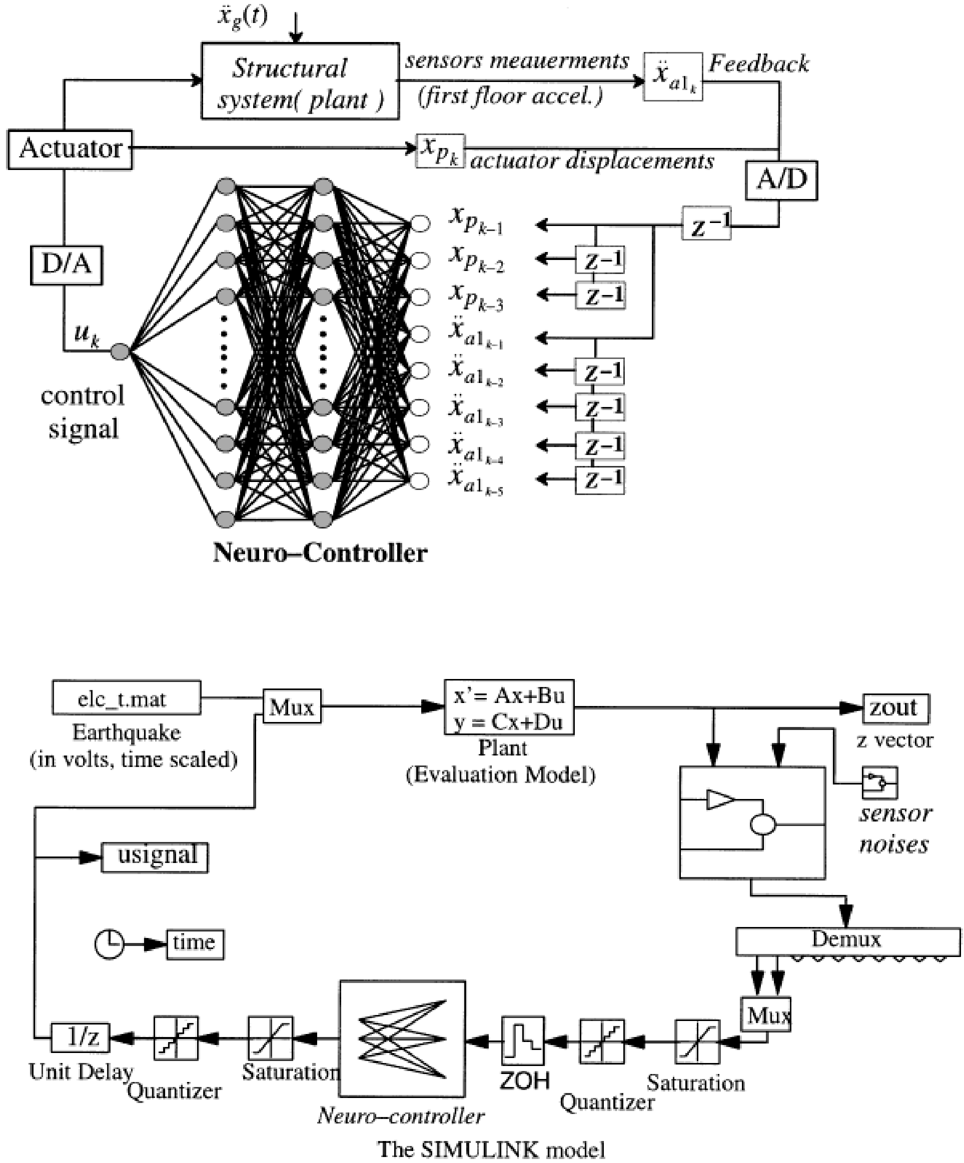


Figure 1. The neuro-controller and its SIMULINK model

Note that the arguments of the function include a portion of the past history of the sensor readings, actuator signal and loading. The constraint equation is a function of the future values of the output vector. The main parameters of this function are the extent of the past digital values of the arguments ℓ, m, n and p. Such a relationship would always exist for sufficiently large values of these parameters. These parameters are in general related to the degree of non-linearity of underlying process represented by the function. Currently, there are no rigorous methods of determining these parameters. It is important to note that the function in equation (1), which must be learned by the neuro-controller also includes the effects of actuator dynamic, actuator saturation, time delays and the sampling period. It is therefore, a highly non-linear function. In designing neuro-controllers, the values of the parameters ℓ, m, n and p are determined by trial and error.

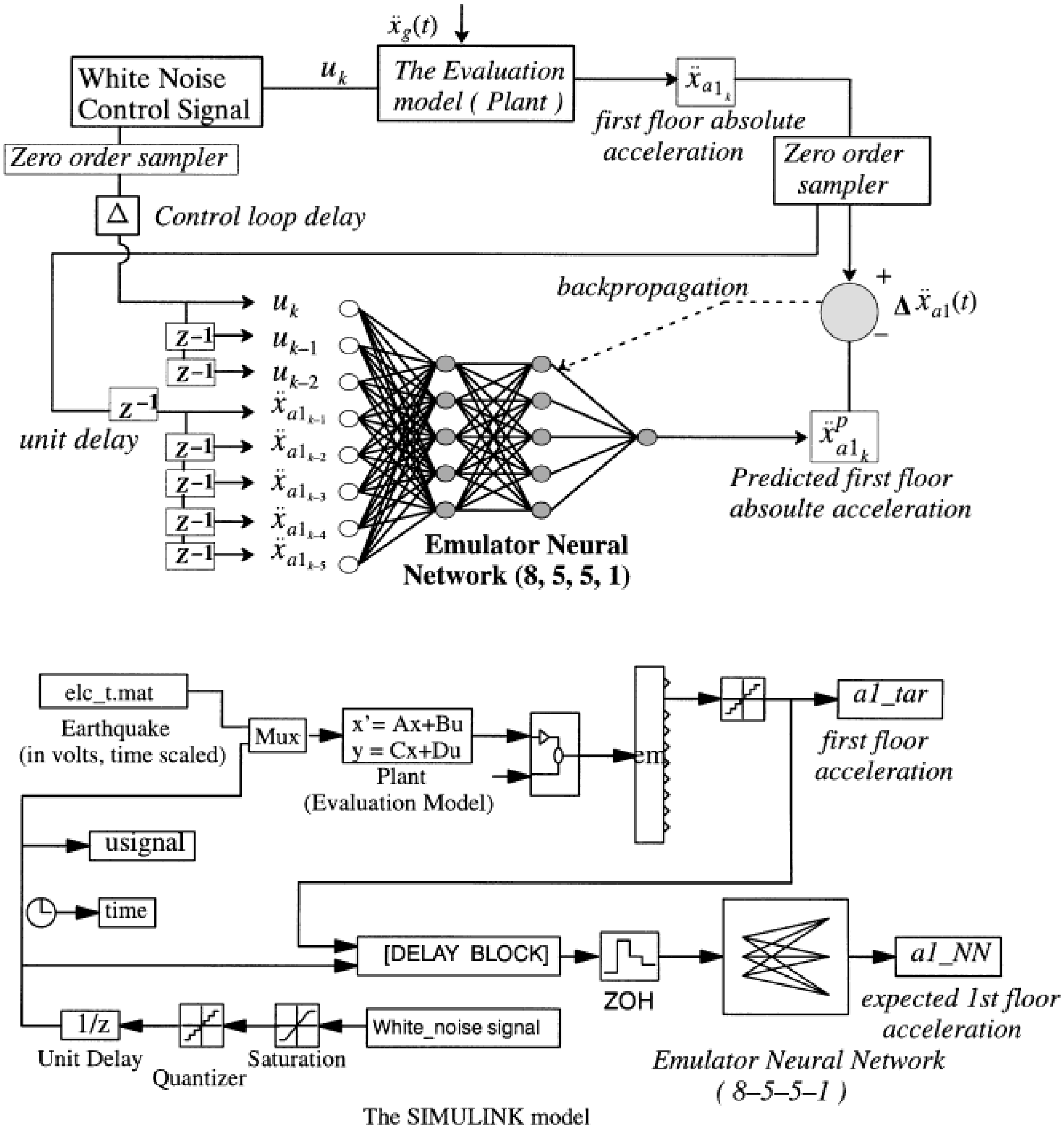


Figure 2. The emulator neural network and its method of training, and The SIMULINK model for the emulator data generation and evaluating

The emulator neural network also learns a relationship represented by the following equation:

$$x_{k+1} = f_e(x_k, x_{k-1}, \ldots, x_{k-r}, u_{k+1}, u_k, u_{k-1}, \ldots, u_{k-s}) \tag{2}$$

Note that this function relates the sensor readings at $k+1$ to the actuator signal at $k+1$ and a portion of the history of the sensor readings and the actuator signal. It is assumed that this relationship uniquely exists for sufficiently large values of the parameters r and s. This equation represents the transfer function between the actuator signal and the sensor readings. Even if the structure itself remains linear, the effects of the actuator dynamics, actuator saturation and the sampling period, which are also included, make this function non-linear. Again, values of the parameters, r and s depend on the degree of non-linearity of the system.

The neuro-controller and the emulator neural network learn the relationships represented by equations (1) and (2). However, neural network reprehension is not exactly the same as the functions they learn. For this reason we use a different symbol to represent the trained neural networks:

$$u_{k+1} = NN_{\mathrm{nc}}(y_k, y_{k-1}, \ldots, y_{k-l}, u_k, u_{k-1}, \ldots, u_{k-m}, p_k, p_{k-1}, \ldots, p_{k-n}) \tag{3}$$

$$x_{k+1} = NN_{\mathrm{e}}(x_k, x_{k-1}, \ldots, x_{k-r}, u_{k+1}, u_k, u_{k-1}, \ldots, u_{k-s}) \tag{4}$$

Whereas, the mathematical functions are exact and universally true, the neural networks approximate these underlying functions over a limited range of interest. The uniqueness requirements for the neural networks are far more relaxed than for mathematical functions. For the neural network training to be successful, the underlying function must exist but need not be strictly unique. Moreover, even if the underlying function uniquely exists, the neural network architecture is not unique; more than one neural network can learn the same underlying function to within a given degree of accuracy over a limited range.

In training of any neural network, a set of training cases, consisting of input/output pairs, are needed. The training cases for the emulator neural network are generated either through numerical simulation (the *evaluation* model and the SIMULINK[11] program in our case) or in an experimental setting by sending signals to the actuator and recording the sensor outputs. The same procedure cannot be used for generating training cases for the neuro-controller, since the correct values of the output are not known. The purpose of the emulator neural network is to provide a path for back-propagation of the errors in training of the neuro-controller. The procedure for training of the neuro-controller with the aid of the emulator neural network is schematically shown in Figure 3.

The neural network training method used in this study, as well as the previous studies by Ghaboussi and his co-workers, is an adaptive architecture determination method, which was originally developed in 1990[13–15] and, has since been modified and improved.[16] This method, combines the 'Quickprop' training algorithm proposed by Fahlman[17] and the dynamic node generation method proposed by Ash.[18] The essentials of the training method used in this study has been described in Joghataie, Ghaboussi and Wu.[16] The training process starts with a small part of the training cases and small number of nodes in the hidden layers (not less than two nodes). As the training proceeds, more training cases are added and the rate of learning is monitored. When the rate of learning falls below a certain value, which indicates that the network is approaching its capacity, one new node is added to each hidden layer. The training is continued for a time with the old connection weights frozen, so that the new connection weights can learn part of the knowledge which was not learned by the old connection weights. Subsequently, the old connection weights are unfrozen and the training continues with all the connection weights. This process is continued until all the neural network learn all the training cases.

The adaptive training method which was briefly described in the previous paragraph, obviously has many parameters, such as: how to divide the training cases into smaller packets; when to add a new packet of training cases; why add one node at a time to the hidden layers instead of two or more nodes; how long should the old connection weight be frozen, etc. As in many other aspects of the neural networks, there are no unique answers to these questions. Our experience has shown that the overall training of the neural networks and the performance of the trained networks are, to a great extent, insensitive to these parameters. We have developed a set a empirical rules which work for these class of problems. Some of these rules have been described in Reference 16. However, the relative effectiveness of these rules may be problem dependent.

BENCHMARK PROBLEM, ACTIVE TENDON SYSTEM

The system considered in this study is a 1:4 scale model of a three-storey building considered in Chung *et al.*,[19] that has become a standard model for structural control problems. Several experiments have been performed on this model at NCEER at SUNY, Buffalo. A similar system has been built in the Department of Civil Engineering at the University of Illinois at Urbana Champaign by Ghaboussi and his co-workers, for

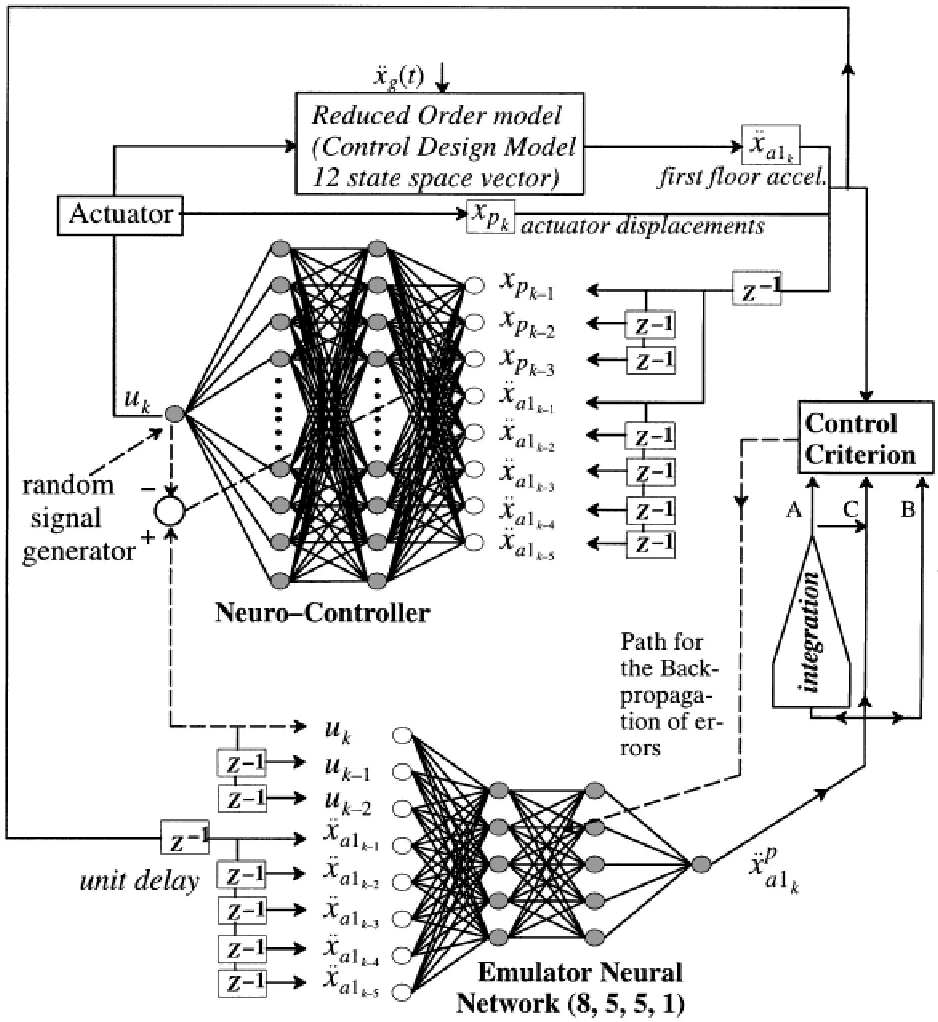


Figure 3. Schematics of the training of the neuro-controller with the aid of the emulator neural network

testing and evaluating the capabilities of the neuro-controllers in linear and non-linear structural control. Control system consists of a hydraulic control actuator and a tendon/pulley system. The frame has a total mass of 2950 kg, distributed evenly among the three floors, and a height of 254 cm. The structure has three distinct, lightly damped modes with the natural frequency values of 2·33, 7·37 and 12·24 Hz, and damping ratios of 0·6, 0·7, and 0·4%. The structure was fully monitored, but the only practical measurements that can be used in the control feedback are the accelerations and the actuator displacements.

Mathematical simulation (*evaluation model*)

The *evaluation* model is considered to be the true mathematical model of the control system considering the interaction among all of its components (structure, actuator, sensors and tendons). However, the model representation is assumed to be accurate up to 50 Hz. The *evaluation* model described here, is the same model provided for the benchmark problem. The model has 20 states and can be described mathematically

as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) + \mathbf{E}\ddot{x}_{\mathrm{g}} \tag{5}$$

$$\mathbf{y}(t) = \mathbf{C}_y\mathbf{x}(t) + \mathbf{D}_y\,u(t) + \mathbf{F}_y\ddot{x}_{\mathrm{g}} + \mathbf{v}(t) \tag{6}$$

$$\mathbf{z}(t) = \mathbf{C}_z\mathbf{x}(t) + \mathbf{D}_z\,u(t) + \mathbf{F}_z\ddot{x}_{\mathrm{g}} \tag{7}$$

where, $\mathbf{x}(t)$ is the 20-dimensional state-space vector, $u(t)$ the single actuator signal, $\ddot{x}_{\mathrm{g}}$ the horizontal ground acceleration, $\mathbf{v}(t)$ the measurement noise vector, $\mathbf{y}(t) = [x_{\mathrm{p}}, \ddot{x}_{\mathrm{a}1}, \ddot{x}_{\mathrm{a}2}, \ddot{x}_{\mathrm{a}3}, f, \ddot{x}_{\mathrm{g}}]^{\mathrm{T}}$ is six-dimensional states observation vector (sensors readings), and $\mathbf{z}(t) = [x_1, x_2, x_3, x_{\mathrm{p}}, \dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_{\mathrm{p}}, \ddot{x}_{\mathrm{a}1}, \ddot{x}_{\mathrm{a}2}, \ddot{x}_{\mathrm{a}3}, f\,]^{\mathrm{T}}$ is the output vector of the system that can be regulated (12 states), and $\mathbf{A}, \mathbf{B}, \mathbf{E}, \mathbf{C}_y, \mathbf{D}_y, \mathbf{F}_y, \mathbf{C}_z, \mathbf{D}_z$ and $\mathbf{F}_z$ are the appropriate matrices. This *evaluation* model has been used in training the emulator and in the evaluation of the neuro-controllers. However, the neuro-controller has been designed using a *reduced order* model, that has been developed with a 12-dimensional state-space vector ($\mathbf{x}^{\mathrm{c}}, \mathbf{x}_r \leqslant 12$) as required by the benchmark problem specifications, although neuro-controllers can be easily trained on the evaluation model itself and they do not require a reduced-order model.

The emulator neural network has some advantages over the classical identification methods; it can be trained with the measured data during an experiment or from the recorded time histories, without requiring any mathematical formulation. As a result, we can use the *evaluation* model with its 20-dimensional state-space vector to train the emulator neural network, assuming that the *evaluation* model represents the actual structure. The emulator neural network can be described in the following form:

$$\ddot{x}_{\mathrm{a}1_k} = NN_{\mathrm{e}}(\ddot{x}_{\mathrm{a}1_{k-1}}, \ \ldots, \ddot{x}_{\mathrm{a}1_{k-5}}, u_{k-2}, u_{k-1}, u_k) \tag{8}$$

where $\ddot{x}_{\mathrm{a}1_k}$ and u_k are the first floor absolute acceleration and the control signal at $t = kT_{\mathrm{s}}$. Obviously, the emulator can predict the response of the structure from the history of the responses, and the current signal and a portion of the past history of the signal. On the other hand, the neuro-controller, is trained using the *reduced-order* model and the emulator neural network. The 12 state *reduced-order* model has been developed using the balanced model truncation technique with the tools found in the MATLAB[20] control toolbox. Although the neuro-controller is trained on the *reduced-order* model, when the trained neuro-controller is used it receives its feedback from the actual structure (*evaluation model*). The neuro-controller can be written in the following form:

$$u_k = NN_{\mathrm{nc}}(\ddot{x}_{\mathrm{a}1_{k-1}}, \ \ldots, \ddot{x}_{\mathrm{a}1_{k-5}}, x_{\mathrm{p}_{k-1}}, x_{\mathrm{p}_{k-2}}, x_{\mathrm{p}_{k-3}}) \tag{9}$$

where $\ddot{x}_{\mathrm{a}1_k}$, x_{p_k} are the first floor absolute acceleration and the actuator displacement at $t = kT_{\mathrm{s}}$.

The simulation of the system dynamics was done using the SIMULINK[11] software with the following simulation parameters: sampling time period $T_{\mathrm{s}} = 0{\cdot}001$ sec, control loop time delay $\tau = 200$ μsec, integration time step $dt = 0{\cdot}0001$ sec. The A/D, D/A converters on the digital control have a 12 bit precision and a span of ± 3V, which gives a resolution of $\approx 1{\cdot}5$ mV. These values have been stated explicitly in the benchmark problem.

Performance indexes, and the evaluations criteria

Some common evaluation criteria have been selected, to be used in the benchmark problem, so the comparison can be made. These criteria are divided into two categories.

Performance criteria based on the rms of the responses

The first set of performance criteria, are the values of the root mean squares (rms) of the structural responses when the system is subjected to a stationary random process with the spectral density function defined by the Kanai–Tajimi spectrum.

$$S_{\ddot{x}_{\mathrm{g}}\ddot{x}_{\mathrm{g}}}(w) = \frac{S_o(4\zeta_{\mathrm{g}}^2\,w_{\mathrm{g}}^2\,w^2 + w_{\mathrm{g}}^4)}{(w^2 - w_{\mathrm{g}}^2)^2 + 4\zeta_{\mathrm{g}}^2\,w_{\mathrm{g}}^2\,w^2} \tag{10}$$

where it is assumed that w_g and ζ_g vary within the ranges of: $8\text{rad}/s \leqslant w_g \leqslant 50$ rad/sec and $0{\cdot}3 \leqslant \zeta_g \leqslant 0{\cdot}75$. The spectral intensity S_0 is chosen such that the rms of the ground motion is constant and has a value of 0·034 g. The following performance indexes will to be evaluated and reported:

$$J_1 = \max_{w_g, \zeta_g} \left\{ \frac{\sigma_{d_1}}{\sigma_{x_{30}}}, \frac{\sigma_{d_2}}{\sigma_{x_{30}}}, \frac{\sigma_{d_3}}{\sigma_{x_{30}}} \right\} \tag{11}$$

$$J_1 = \max_{w_g, \zeta_g} \left\{ \frac{\sigma_{\ddot{x}_{a1}}}{\sigma_{\ddot{x}_{a30}}}, \frac{\sigma_{\ddot{x}_{a2}}}{\sigma_{\ddot{x}_{a30}}}, \frac{\sigma_{\ddot{x}_{a3}}}{\sigma_{\ddot{x}_{a30}}} \right\} \tag{12}$$

$$J_3 = \max_{w_g, \zeta_g} \left\{ \frac{\sigma_{x_p}}{\sigma_{x_{30}}} \right\} \tag{13}$$

$$J_4 = \max_{w_g, \zeta_g} \left\{ \frac{\sigma_{\dot{x}_p}}{\sigma_{\dot{x}_{30}}} \right\} \tag{14}$$

$$J_5 = \max_{w_g, \zeta_g} \left\{ \frac{\sigma_f}{W} \right\} \tag{15}$$

where σ_{d_i} is the rms inter-storey drift value for the ith floor, $\sigma_{\ddot{x}_{ai}}$ the rms value for the absolute acceleration of the ith floor, σ_{x_p} the rms value for the actuator displacements, $\sigma_{\dot{x}_p}$ the rms value for the actuator velocity, σ_f the rms value for the actuator force, and W the total structure weight (= 289 kN). The maximum rms displacement for the third floor of the uncontrolled case were found to be $\sigma_{x_{30}} = 2{\cdot}34$ cm, the maximum uncontrolled third floor velocity $\sigma_{\dot{x}_{30}} = 33{\cdot}3$ cm/sec, and the maximum uncontrolled third floor absolute acceleration $\sigma_{\ddot{x}_{a30}} = 0{\cdot}485$ g. These values occur when $w_g = 14{\cdot}5$ rad/s and $\zeta_g = 0{\cdot}3$. Three other hard constraints were imposed for the neuro-controller: the rms of the control signal $\sigma_u \leqslant 1{\cdot}0$ V; the rms of the control force $\sigma_f \leqslant 4{\cdot}0$ kN; and, the rms of the actuator displacement $\sigma_{x_p} \leqslant 1{\cdot}0$ cm.

Performance criteria based on the peak responses

The second set of performance criteria are based on the peak responses of the controlled system, when the system is subjected to two compressed earthquake records; the 1940 E1 Centro NS record and the 1968 Hachinohe NS record. The following performance indexes will be evaluated and reported:

$$J_6 = \max_{\substack{\text{Hachinohe} \\ \text{El Centro}}} \left(\max_{(t)} \left\{ \frac{|d_1|}{x_{30}}, \frac{|d_2|}{x_{30}}, \frac{|d_3|}{x_{30}} \right\} \right) \tag{16}$$

$$J_7 = \max_{\substack{\text{Hachinohe} \\ \text{El Centro}}} \left(\max_{(t)} \left\{ \frac{|\ddot{x}_{a1}|}{\ddot{x}_{a30}}, \frac{|\ddot{x}_{a2}|}{\ddot{x}_{a30}}, \frac{|\ddot{x}_{a3}|}{\ddot{x}_{a30}} \right\} \right) \tag{17}$$

$$J_8 = \max_{\substack{\text{Hachinohe} \\ \text{El Centro}}} \left(\max_{(t)} \left\{ \frac{|x_p|}{x_{30}} \right\} \right) \tag{18}$$

$$J_9 = \max_{\substack{\text{Hachinohe} \\ \text{El Centro}}} \left(\max_{(t)} \left\{ \frac{|\dot{x}_p|}{\dot{x}_{30}} \right\} \right) \tag{19}$$

$$J_{10} = \max_{\substack{\text{Hachinohe} \\ \text{El Centro}}} \left(\max_{(t)} \left\{ \frac{|f|}{W} \right\} \right) \tag{20}$$

where d_i is the inter-storey drift for the ith floor, $\ddot{x}_{ai}$ is absolute acceleration of the ith floor, x_p is the actuator piston displacement, $\dot{x}_p$ is the actuator piston velocity, f is the actuator force, and W is the total structure

weight ($= 289$ kN). For the uncontrolled system subjected to El Centro Earthquake we have the following: maximum displacement at the third floor $x_{30} = 6{\cdot}45$ cm; maximum velocity at the third $\dot{x}_{30} = 99{\cdot}9$ cm/sec; and, maximum absolute acceleration at the third floor $\ddot{x}_{30} = 1{\cdot}57g$. Similarly when the system is subjected to Hachinohe earthquake record we have the following maxima: $x_{3_o} = 3{\cdot}78$ cm; $\dot{x}_{3_o} = 56{\cdot}1$cm/sec; and, $\ddot{x}_{3_o} = 0{\cdot}778$g. The hard constraints imposed on the neuro-controller were: the absolute maximum control signal should not exceed 3·0V; the absolute maximum control force should not exceed 12·0 kN; and, the absolute maximum actuator displacement should not exceed 3·0cm.

CONTROLLER DESIGN

The controller design using neural network methodology, has been developed and tested by Ghaboussi and his co-workers in previous works, where two different methods were presented. A neuro-controller based on the inverse transfer function was introduced and implemented experimentally by, Nikzad *et al.*[7] In a second method introduced by Ghaboussi and his co-workers,[1,2,4] also used in this study, they used an emulator neural network to train the neuro-controller. The neuro-controller design in this method, can be divided into two parts: first the emulator neural network is trained and evaluated; then, the neuro-controller is trained on line with the aid of the emulator neural network. In this study one emulator neural network was developed and it was used to develop three different neuro-controllers, each with a different control criterion.

Emulator neural network (*neuro-identifier*)

The emulator is the first neural network to be trained. The emulator learns to predict the response of the structure from the immediate past history of the structural response. The emulator is chosen to have two hidden layers. The input layer consists of eight nodes which represent: the absolute acceleration of the first floor of the frame at the last five past sampling periods; the current actuator signal and the actuator signals at the past two sampling periods. The single output node represents the current absolute acceleration of the first floor. Figure 2 shows the architecture of the emulator neural network as well as its method of training. The SIMULINK[11] program, used in preparing the training data for the emulator, as well as in evaluating its performance, is also shown in Figure 2.

The emulator neural network can be considered a black box which represents the transfer function between the actuator signal and the sensor readings. Clearly, this transfer function includes the structural behaviour. Therefore, it can be stated that the emulator learns some part of the structural behavior. However, the transfer function also includes the effects of the actuator dynamics and sampling period. The emulator learns to incorporate the effects of these important factors. The neuro-controller also learns to compensate for the effects of the actuator dynamics, sampling period and the control loop time delays when it is trained with the aid of the emulator neural network.

Training of the emulator neural network has been accomplished by using the SIMULINK[11] program provided in the benchmark problem. Three analyses have been performed using the 20 state *evaluation model*: in the first analysis the system was subjected to the compressed 1940 El Centro earthquake NS record while the control force was turned off; in the second analysis the system was subjected to random white noise actuator signal with no earthquake input; finally, in the third analysis the system was subjected to a combination of the earthquake ground motion and the white noise actuator signal. A 10·0 sec portion of the results from three analyses was used to generate a total of 30000 training patterns for the emulator neural network. Figure 2 shows the SIMULINK[11] model used in preparing the training data for the emulator neural network. The same SIMULINK[11] model was also used in evaluating the performance of the trained emulator by adding the neural network block.

The performance of the emulator neural network was evaluated by comparing its response with the results of the analysis by the SIMULINK[11] program using the 20 state *evaluation* model. This evaluation was performed for three different cases: (1) 100 sec period of white noise actuator command; (2) 100 per cent of

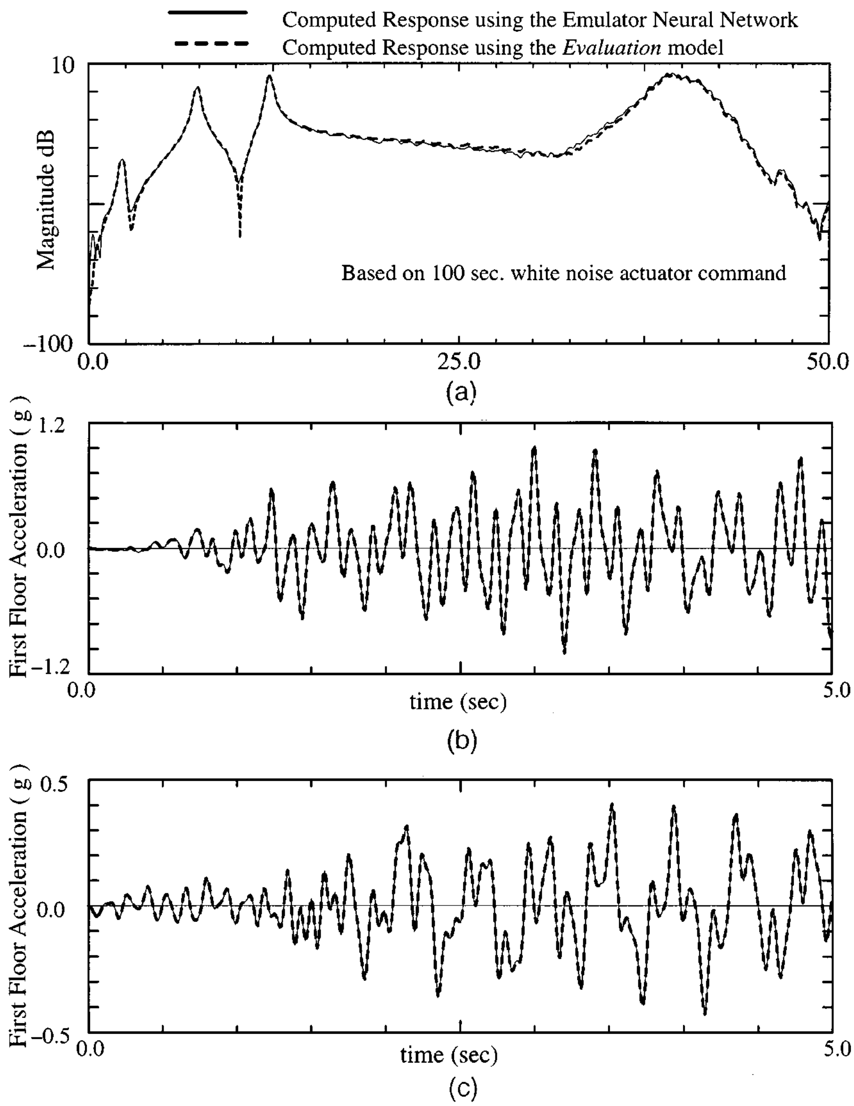


Figure 4. Comparison of the neural network emulator response and the computed first floor response of the structure using the *evaluation* model in the SIMULINK program: (a) the transfer function from the command signal to the first floor acceleration; (b) the structure is subjected to 100% of 1940 El centro NS earthquake record; (c) the structure is subjected to 100% of 1968 Hachinohe NS earthquake record

1940 El Centro earthquake record; (3) 100 per cent of 1968 Hachinohe earthquake record. The results are shown in Figure 4. Clearly, the emulator neural network has been able to learn the transfer function from the control command to the first floor acceleration very well, and to reproduce the structural response under different excitations very accurately.

Neuro-controller

The methodology for the training of the neuro-controller used in this study is also described in Reference 1. The method is based on using the emulator neural network on-line to develop the training data set for the controller. The procedure for training of the neuro-controllers is shown in Figure 3. A random control signal is sent to the actuator, while the structure is subjected to the base motion. The structural response at one

sampling period is collected and sent to the box labeled the control criterion. Simultaneously, the control signal is fed to the emulator neural network and its output is also sent to the box labeled the control criterion. In the control criterion box, the control error is determined and backpropagated through the emulator neural network and through the neuro-controller. Only the connection weights of the neuro-controller are modified. This process is repeated until the control criterion is satisfied and the control error is reduced to below a specified tolerance.

Mathematically, the process of backpropagating the control error through the emulator neural network, can be approached in different ways. Ghaboussi and Joghataie[4] used an iterative loop for calculating the control signal which satisfies the control criterion. They first computed the Jacobian, representing the sensitivity of the acceleration of the first floor with respect to the actuator signal. Then, they used the inverse of the Jacobian to calculate a correction for the actuator signal. When this process is applied iteratively, it leads to a control signal which will either satisfy the control criterion, or will cause the actuator saturation. Chen *et al.*,[10] made use of the internal architecture of the emulator. They backpropagated the control error through the emulator to determine the differential actuator signal error at the input of the emulator neural network. This scheme is repeated continuously, until the control criterion is satisfied for every sampling period or the actuator reaches saturation.

In this study, we use a method similar to the one used in Ghaboussi and Joghataie.[4] It can be described as a search method. For each time step, we search for the actuator signal which satisfies the control criterion. This search is conducted by alternately varying the value of the actuator signal, $u_j, j = 1, \ldots, n$, between zero and its limits $u_{\max}$and u_{min} (-3 and 3V in our case) by increments of Δu_j. The total number of increments n is determined by the following equation:

$$n = \left[\frac{u_{\max} - u_{\min}}{\Delta u} \right] \tag{21}$$

When $j = 0$ the case is called the *uncontrolled reference case*. The training set for the neuro-controller at each time step is obtained by satisfying the control criterion.

For each actuator signal increment, the emulator neural network is used to predict the structural response at m future time steps. The actuator signal u_j is assumed to remain constant over the next m time steps while the predicted structural response $\ddot{x}^{pi}_{aj}$, $i = 1, \ldots, m$ is determined using the emulator neural network. The control criterion is then based on a time-averaged value of the structural response. However, since the reliability of the emulator predicted structural response deteriorates with elapsed time, a weight function is assigned to each predicted response value. The weight function, called the *importance function* and it is a decreasing function of time.

Control criteria

Three neuro-controllers have been designed. The neuro-controller A has been designed with a control criterion based on the reduction of the predicted integrated relative displacements of the first floor. Neuro-controller B has been designed with a control criterion based on the reduction of the first floor accelaration. Finally, neuro-controller C has been designed with a control criterion based on the simultaneous reduction of the first floor relative displacements and first floor accelaration.

For neuro-controller A, the predicted absolute acceleration of the first floor is integrated twice to determine the relative displacement, using Wilson's Θ method.[21] Wilson's Θ method, uses the following relations between the accelerations, velocities and displacements at two successive time steps:

$$x^{i}_{1_j} = x^{i-1}_{1_j} + C_1 \Delta x^{i}_{1_j} + C_2 \dot{x}^{i-1}_{1_j} + C_3 (\ddot{x}^{i-1}_{a1_j} - \ddot{x}_g) \tag{22}$$

$$x^{i}_{1_j} = C_4 \Delta x^{i}_{1_j} + C_5 \dot{x}^{i-1}_{1_j} + C_6 (\ddot{x}^{i-1}_{a1_j} - \ddot{x}_g) \tag{23}$$

$$\ddot{x}^{i}_{a1_j} = C_7 \Delta x^{i}_{1_j} + C_8 \dot{x}^{i-1}_{1_j} + C_9 (\ddot{x}^{i-1}_{a1_j} - \ddot{x}_g) + \ddot{x}_g \tag{24}$$

where i indicates the predicted time step and j is the current actuator signal increment step and, $C_1 - C_9$ are functions of integration constant $\Theta \geqslant 1{\cdot}4$ and the integration time step Δt. By rearranging Equation

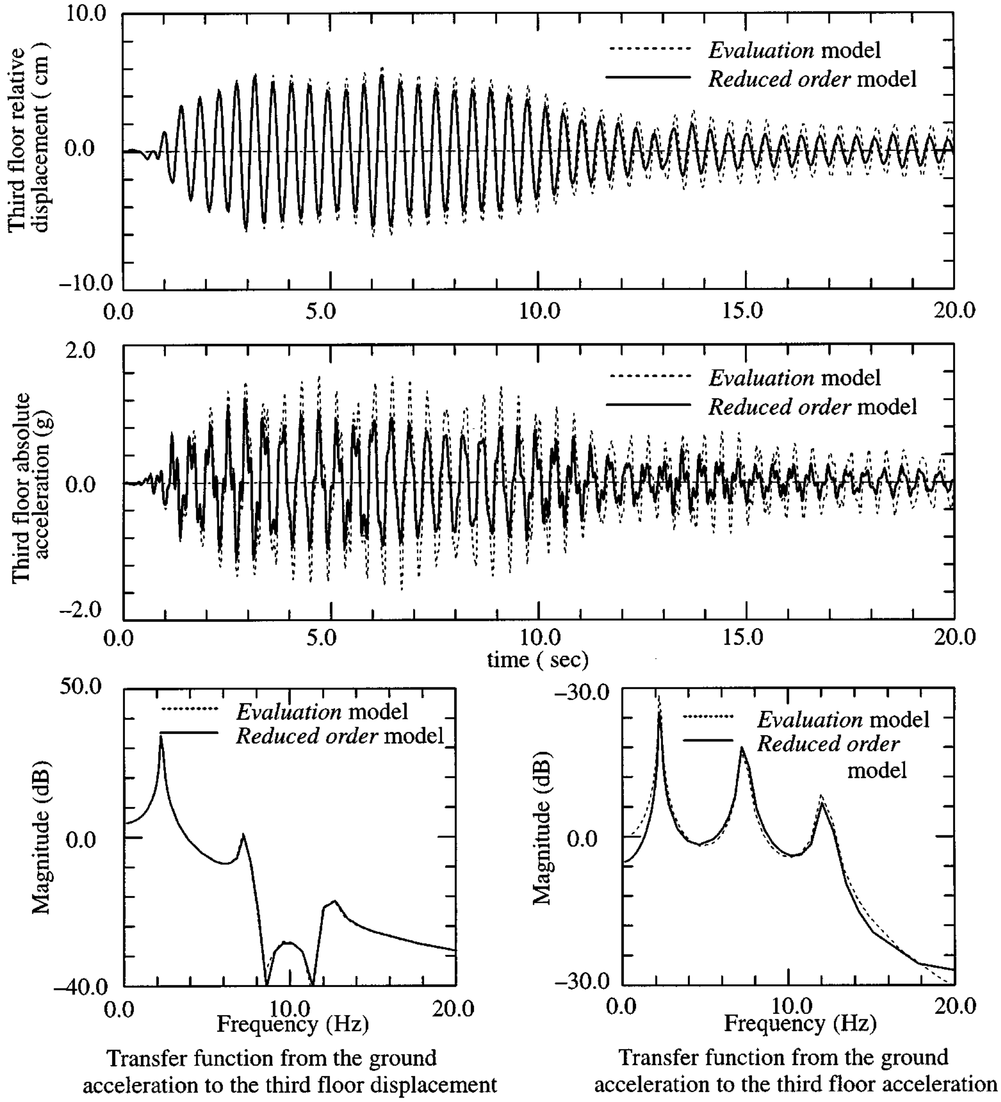


Figure 5. Comparison between the responses of the structure using the *evaluation* model and the *reduced-order* model (design model), when the system is subjected to El Centro earthquake record

(22)–(24), the current values of the relative displacement and the relative velocity of the first floor can be determined in terms of the current value of the absolute acceleration, and the values of the relative displacement, relative velocity and relative acceleration at the previous time step.

Assuming that the predicted accelerations are reasonably accurate, the relative displacement can be estimated for several future time steps and used in the control criterion for neuro-controller A.

An equivalent first floor relative displacement $\bar{X}_j$ is computed by using the appropriate importance function W_i, and the displacements integrated from the predicted accelerations.

$$\bar{X}_j = \left\{ \frac{\sum_{i=0}^{m} |x_{1_j}^i| * W_i}{\sum_{i=0}^{m} W_i} \right\} \tag{25}$$

The average reference value for the uncontrolled case is $\bar{X}_{\mathrm{R}} = \bar{X}_0 * C_{\mathrm{d}o}$, for $j = 0$ where $C_{\mathrm{d}o} \leqslant 1{\cdot}0$ is a reduction factor. The control signal is chosen to satisfy the following control criterion: $\bar{X}_j \leqslant \bar{X}_{\mathrm{R}}$ and $\bar{X}_j \leqslant \varepsilon_{\mathrm{d}}$, where ε_{d} is the control tolerance. If this criterion can not be met, then the control signal is chosen to minimize $\bar{X}_j$. The numerical values of the parameters used in the control criterion for training of the neuro-controller A are: $m = 4$, $\Delta u = 0{\cdot}001\mathrm{V}$, $n = 6000$, $\varepsilon_{\mathrm{d}} = 0{\cdot}2$ cm, $C_{\mathrm{d}o} = 0{\cdot}9$ and the importance function is defined by the following equation:

$$W_i = 1{\cdot}5 - \frac{(i-1)}{m}, \quad i = 1, \ldots, m \tag{26}$$

For the neuro-controller B, no integration was necessary since the control criterion is based on the reduction of the first floor absolute acceleration. An equivalent first floor acceleration $\bar{\ddot{X}}_j$ has been estimated with appropriate importance function W'_i.

$$\bar{\ddot{X}}_j = \left\{ \frac{\sum_{i=0}^{m} |\ddot{x}^{i}_{1\mathrm{a}_j}| * W_i}{\sum_{i=0}^{m} W_i} \right\} \tag{27}$$

The average reference value for the uncontrolled case is $\bar{\ddot{X}}_{\mathrm{R}} = \bar{\ddot{X}}_0 * C_{\mathrm{a}0}$, for $j = 0$, and $C_{\mathrm{a}0} \leqslant 1{\cdot}0$. Similarly, the control criterion is chosen such that the control signal satisfies the following: $\bar{\ddot{X}}_j \leqslant \bar{\ddot{X}}_{\mathrm{R}}$ and $\bar{\ddot{X}}_j \leqslant \varepsilon_{\mathrm{a}}$, where ε_{a} is the control tolerance. If this criterion cannot be met, then the control signal is chosen to minimize $\bar{\ddot{X}}_j$. The numerical values of the parameters used in the control criterion for training of the neuro-controller

Table I. Evaluation Performance Indexes for the three designed controllers, The RMS performance and constraint values were evaluated at the nominal design point $wg = 14{\cdot}5$, $zg = 0{\cdot}3$

Controller A: (Reduction of the first floor relative disablement criterion)								
RMS responses criteria	J_1	J_2	J_3	J_4	J_5	σ_u (V)	σ_{x_p} (cm)	σ_f (kN)
Broadband (K–T)	0·1871	0·3867	0·0396	0·0416	0·0090	0·7020	0·0927	2·6021
Peak responses criteria	J_6	J_7	J_8	J_9	J_{10}	$u_{\max}$ (V)	$x_{p\max}$ (cm)	$f_{\max}$ (kN)
El Centro	0·2743	0·7127	0·0674	0·2540	0·0379	3·9342	0·4345	10·960
Hachinohe	0·3373	0·8460	0·0721	0·1014	0·0238	2·0996	0·2725	6·8805
Controller B: (Reduction of the first floor absolute acceleration criterion)								
RMS responses criteria	J_1	J_2	J_3	J_4	J_5	σ_u (V)	σ_{x_p} (cm)	σ_f (kN)
Broadband (K–T)	0·1541	0·3302	0·0366	0·0346	0·0089	0·6791	0·0856	2·5944
Peak responses criteria	J_6	J_7	J_8	J_9	J_{10}	$u_{\max}$ (V)	$x_{p\max}$ (cm)	$f_{\max}$ (kN)
El Centro	0·2384	0·5148	0·0626	0·0804	0·0364	2·9616	0·4035	10·524
Hachinohe	0·3103	0·8052	0·0622	0·0674	0·0279	1·8234	0·2352	8·0771
Controller C: (Reduction of the first floor absolute acceleration and relative displacement criterion)								
RMS responses criteria	J_1	J_2	J_3	J_4	J_5	σ_u (V)	σ_{x_p} (cm)	σ_f (kN)
Broadband (K–T)	0·1454	0·3121	0·0409	0·0360	0·0087	0·7642	0·0958	2·5207
Peak responses criteria	J_6	J_7	J_8	J_9	J_{10}	$u_{\max}$ (V)	$x_{p\max}$ (cm)	$f_{\max}$ (kN)
El Centro	0·2319	0·5112	0·0519	0·0569	0·0374	2·6844	0·3347	10·810
Hachinohe	0·3011	0·7731	0·0708	0·0708	0·0273	2·1203	0·2677	7·8783

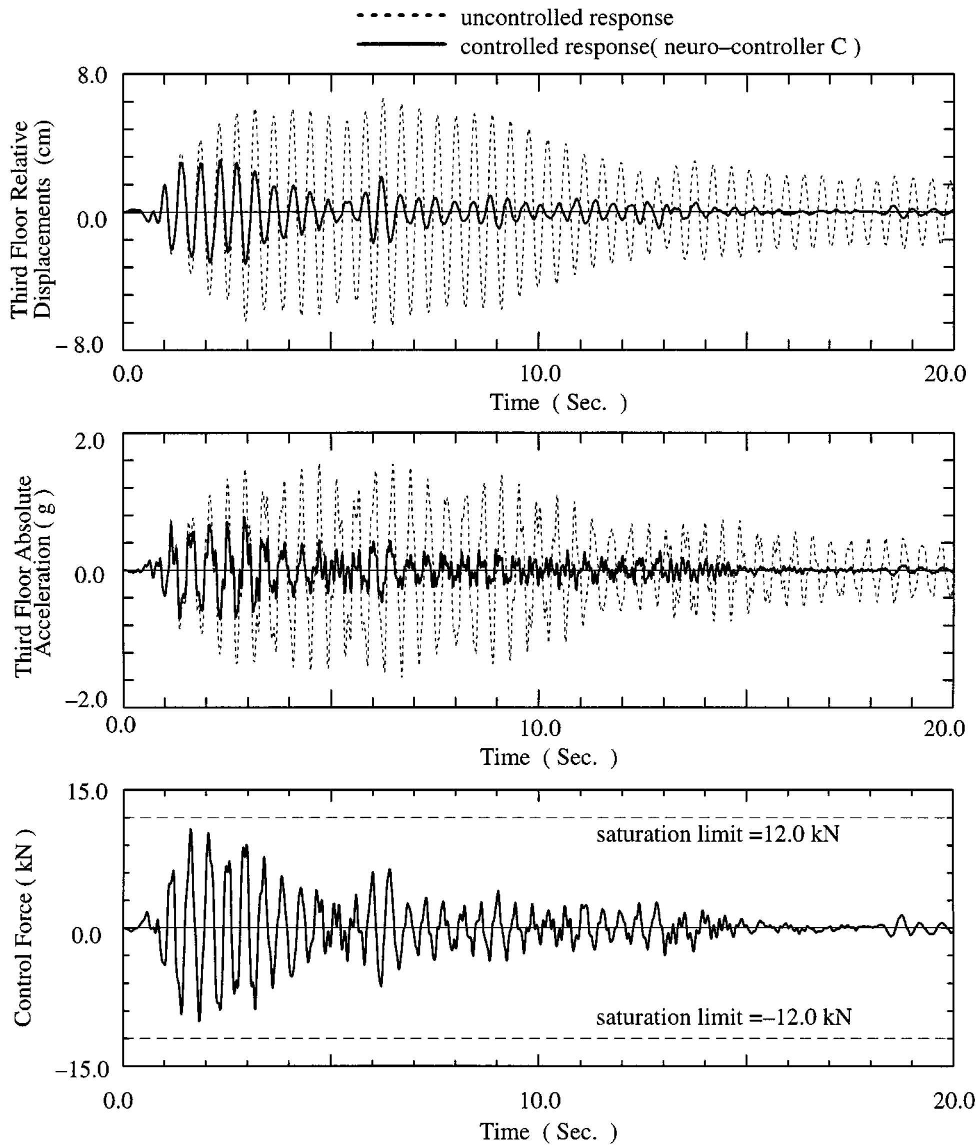


Figure 6. Controlled and uncontrolled responses of the structure subjected to El Centro earthquake record

B are: $m=4$, $\Delta u = 0{\cdot}001$V, $n=6000$, $\varepsilon_a = 0{\cdot}25g$, $C_{a0} = 0{\cdot}95$, and the importance function W_i is defined by equation (26), same as in controller A.

Finally, neuro-controller C, was trained with a control criterion based on the simultaneous reduction of the first floor absolute acceleration and first floor relative displacement. Obviously, this criterion is a combinations of the control criteria for the neuro-controllers A and B. However, the control tolerances ε_d and ε_a were chosen in such a way that more emphasis is placed on the reduction of the first floor acceleration. An equivalent first floor acceleration $\bar{\ddot{X}}_j$ and an equivalent first floor relative displacement $\bar{X}_j$, were computed using appropriate importance functions and equations (25) and (27).

The numerical values of the parameters used in the control criterion for training of the neuro-controller C are: $m=4$, $\Delta u = 0{\cdot}001$ V, $n=6000$, $\varepsilon_d = 0{\cdot}5$ cm, $C_{d0} = 1{\cdot}0$, $\varepsilon_a = 0{\cdot}25g$, $C_{a0} = 0{\cdot}90$, and W_i the same as in equation (26).

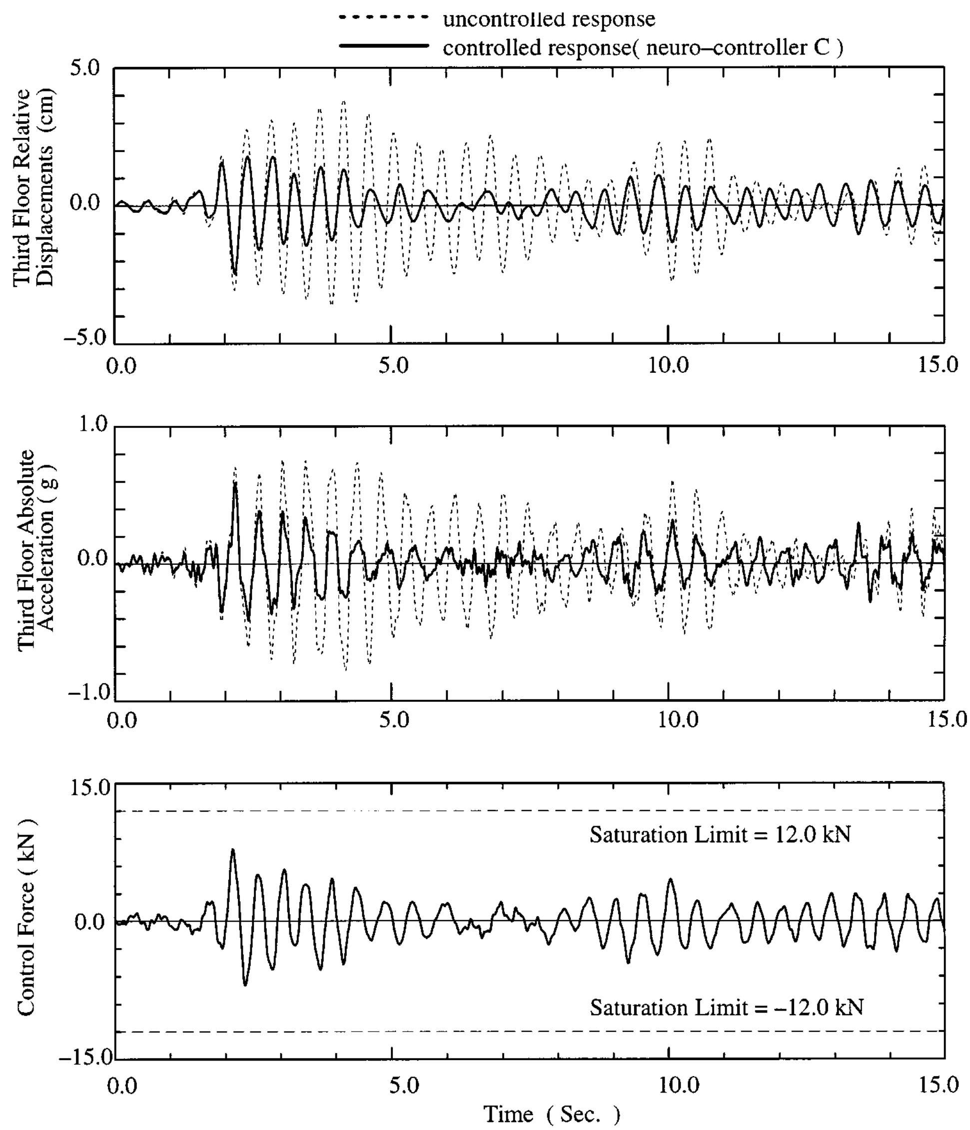


Figure 7. Controlled and uncontrolled response of the structure subjected to Hachinohe earthquake record

The three neuro-controllers A, B, and C were trained using a computer program simulating the methodology shown in Figure 3. The design model for these controllers was a 12 states, *reduced-order* model, which has been developed from the *evaluation* model, using the balanced model truncation technique, available in the MATLAB[20] control toolbox. The *reduced-order* model can be described by the following state-space equations:

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}_r(t) + \mathbf{B}_r u(t) + \mathbf{E}_r \ddot{x}_g \tag{28}$$

$$\mathbf{y}_r(t) = \mathbf{C}_{y_r} \mathbf{x}_r(t) + \mathbf{D}_{y_r} u(t) + \mathbf{F}_{y_r} \ddot{x}_g + \mathbf{v}_r(t) \tag{29}$$

where $\mathbf{x}_r(t)$ is 12 state-space vector, and $\mathbf{y}_r(t)$ is a state vector of the required measurements for the controller design, $\mathbf{y}_r(t) = [x_p, \ddot{x}_{a1}]^T$. Figure 5, shows a comparison between the responses of *evaluation* model and the

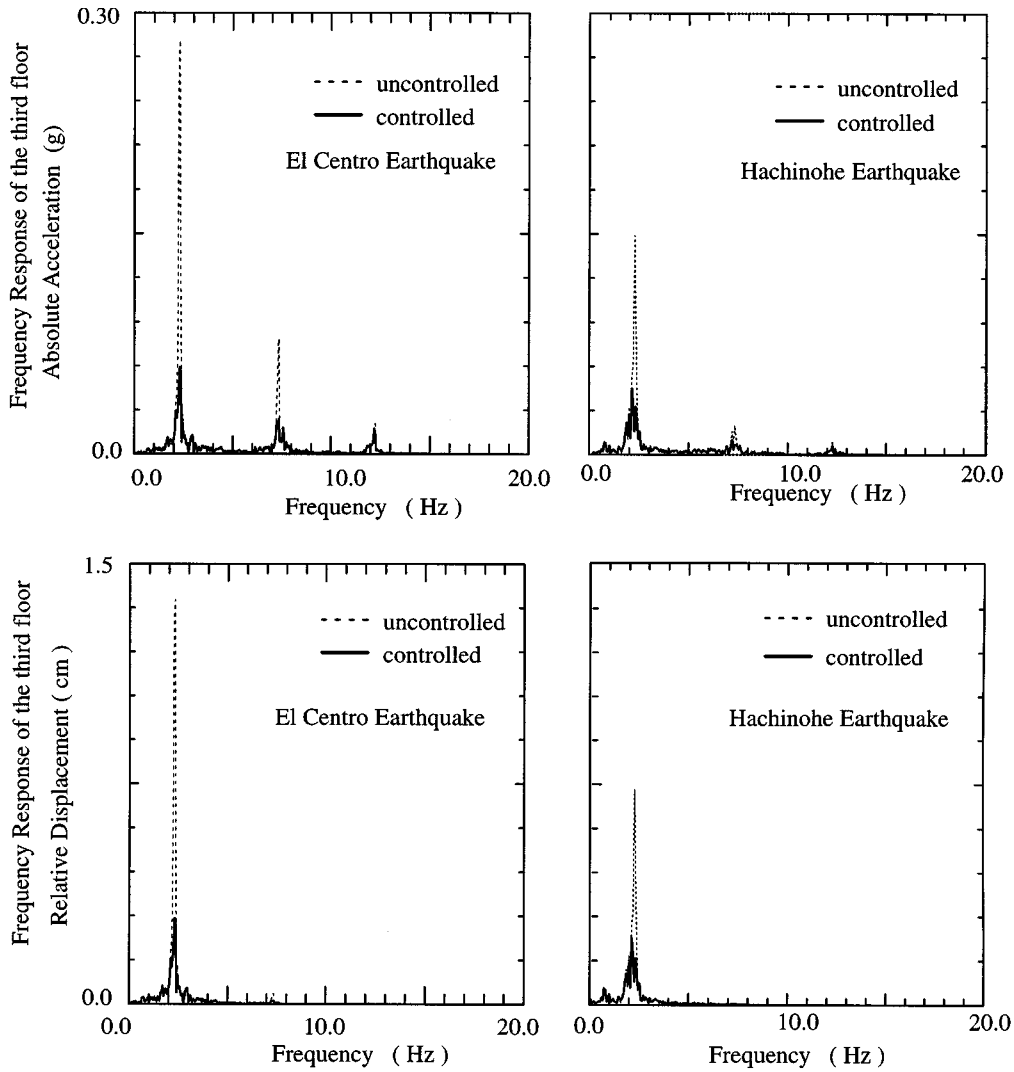
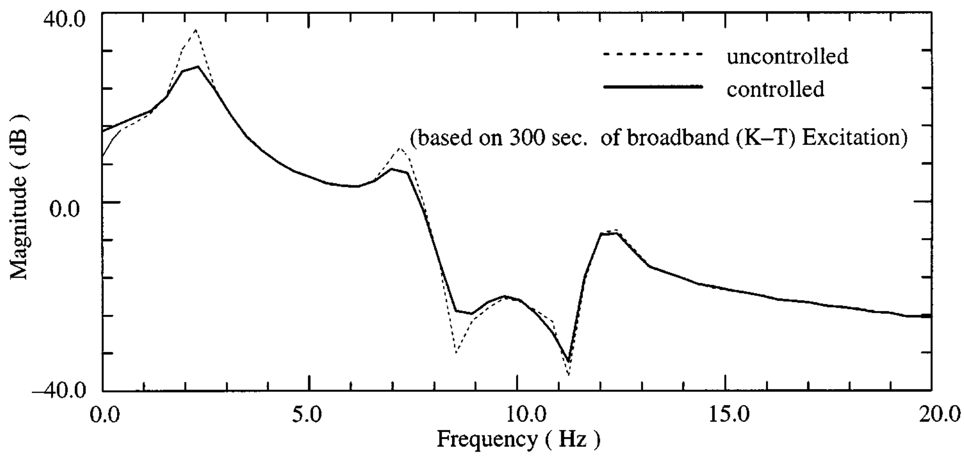
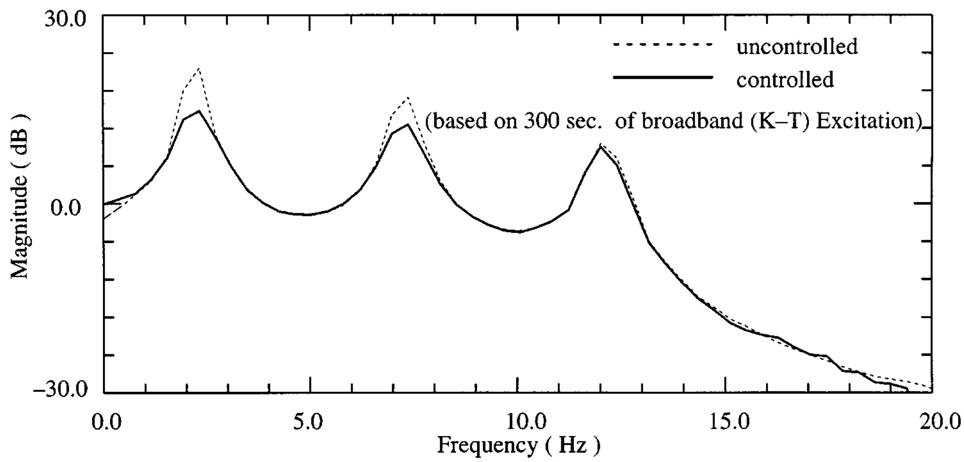


Figure 8. Third floor frequency response for the controlled and uncontrolled structure when the structure is subjected to El Centro earthquake record, and Hachinohe earthquake record (neuro-controller)

reduced-order model in both time domain and frequency domain. It appears that the *reduced-order* model has retained the essence of the original model without significant loss of generalization. In the simulated computer program for the controllers design, both the *reduced-order* model and the emulator neural network were used to develop training cases for the three neuro-controllers. In training the neuro-controllers a 50 per cent amplitude of the compressed 1940 El Centro earthquake record was used for a duration of 10 sec. This generated 10 000 training cases for each neuro-controller. It is interesting to note that, as mentioned earlier, adaptive architecture determination was used and three neuro-controller ended up with different number of node in their hidden layers, adaptively determined during the training process. Neuro-controller A ended the training process with 10 nodes in each of its two hidden layers; neuro-controller B with 7 nodes in each of its two hidden layers; and, neuro-controller C with 6 nodes in each of its two hidden layers. The final number of



(a)



(b)

Figure 9. Comparison between the transfer functions of the ground acceleration to third floor displacement and acceleration, for the controlled and uncontrolled structure. (neuro-controller C). (a) transfer functions from the ground acceleration to the third floor relative displacement; (b) transfer functions from the ground acceleration to the third floor absolute acceleration

nodes in the hidden layers in the adaptive architecture determination in somehow related to the degree of difficulty in learning of the information in the training data set.

NUMERICAL RESULTS

The three neuro-controllers have been evaluated in two stages. In the first stage, the performance of the neuro-controllers were evaluated using the evaluation criteria in equations (11)–(20), as prescribed in the benchmark problem. In the second stage, we study the robustness of the trained neuro-controller by evaluating their performance under severe uncertainties.

The evaluation indexes are: the root mean squares of the controlled responses when the structure is subjected to a stationary random process with the PSD function defined by the Kanai–Tajimi spectrum; and,

Table II. Comparisons between the peak responses of the controlled system and thecontrolled system with some severe uncertainities.

Peak responses of the controlled (neuro-controller C) system subjected to El Centro Earthquake with different severe cases of uncertainities (assessments of robustness and stability)						
Case definition	X_1 (cm)	X_2 (cm)	X_3 (cm)	$\ddot{X}_{a1}$ (g)	$\ddot{X}_{a2}$ (g)	$\ddot{X}_{a3}$ (g)
Uncontrolled system	2·0170	4·9737	6·5653	1·0809	1·2744	1·5631
Evaluation model without any uncertainities	1·0920	2·3080	3·0125	0·6608	0·6267	0·7760
Case 1: Time delay was increased 10 times	1·2125	2·7006	3·6537	0·7116	0·7244	0·9131
Case 2: Uncertainities in the model $\pm$ 15%	1·3643	3·3312	4·3643	0·7412	0·7584	0·9895
Case 3: Uncertainities in sensors readings of $\pm$ 0·3 volts.	1·2147	2·4354	3·3061	0·9378	0·8232	0·8505

the peak controlled responses when the structure is subjected to the compressed El Centro and Hachinohe earthquake records. The neuro-controllers A, B and C, which were trained on 50 per cent amplitude of El Centro earthquake with three different control criteria, were able to control the structure when it is subjected to a more severe earthquake (100 per cent amplitude of El Centro), as well as a different earthquake than the one they were trained on (Hachinohe). Table I summarizes the computed performance indexes for the three neuro-controllers, for the three earthquake excitations. Figure 6 and 7 show the comparison between the controlled and the uncontrolled responses for the third floor absolute acceleration and the third floor absolute acceleration and the third floor relative displacement, for neuro-controller C. Figure 8 shows the same comparisons in the frequency domain.

Clearly, all three neuro-controllers appear effective in controlling the response of the structure. However, comparison of the J_1, J_2, J_6 and J_7 values reveal that neuro-controller C is somewhat more effective than neuro-controllers A and B. It is recalled that neuro-controller C was trained with a control criterion which included both the control criteria used in training neuro-controller A (reduction of first floor relative displacement) and neuro-controller B (reduction of first floor absolute acceleration), with more emphasis placed on the latter criterion. Consequently, we choose the neuro-controller C as our candidate controller, with the performances indexes (0·1454, 0·3121, 0·0409, 0·0360, 0·0087, 0·3011, 0·7731· 0·0708, 0·0708, 0·0374). Figure 9 shows the transfer functions between the ground acceleration and the third floor acceleration and displacement, for the controlled (neuro-controller C) and the uncontrolled system. These transfer functions have been computed using the response of the system when it was subjected to 300 sec of broadband excitation with the K–T spectral density.

The *robustness* of the neuro-controller C is evaluated by computing the uncontrolled and controlled responses of the structure for three different types of uncertainties, introduced by modifying some parameters of the system. These parameter modification are considered unmodelled since the structure was controlled with the original neuro-controller C, trained on the unmodified structure. The first uncertainty represented a type of malfunction which caused a ten fold increase in the time delay. The second uncertainty represented a modification of the structural parameters, possibly caused by damage. It was modelled by modifying the state space parameters of the *evaluation model* by $\pm$ 15 per cent. The third uncertainty simulated the case of a partial sensor failure and, it was modelled by adding a random error of $\pm$ 0·3 V to the sensor feedback. The performance of the neuro-controller C under these unmodelled parameter modifications is summarized in Table II. It is clear that the neuro-controller was still able to perform well, even though the performance is

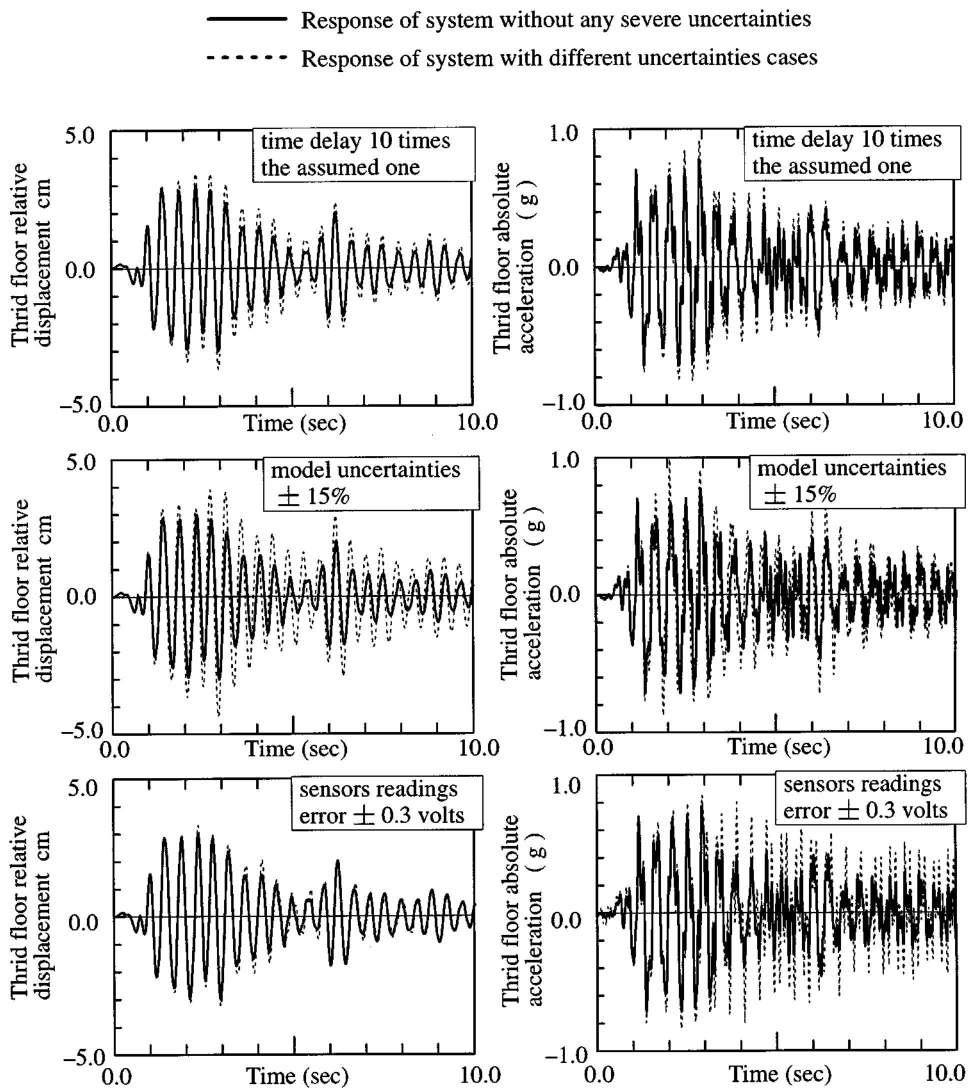


Figure 10. Comparison between the system responses with different cases of uncertainities and the case of the controlled system without uncertainities (neuro-controller C)

somewhat degraded from that of the perfectly modelled case. Figure 10 shows the responses of the system with the three different cases of parameter modifications.

CONCLUDING REMARKS

Three neuro-controllers were designed, trained and evaluated in this study. The results of this study show that a neural network can be successfully implemented in structural control. Neuro-controllers have many advantages over the mathematically formulated control algorithms. While learning to control the structure, they also learn to compensate for the time delays and the actuator dynamics and, they learn to account for

the actuator saturation. We have demonstrated the robustness of the neuro-controllers with results that show their effectiveness, without significant degradations in their performance, under uncertainities represented by unmodelled parameter changes. This study has also demonstrated the effectiveness of the neuro-controllers with minimal feedback, which in this case included the first floor absolute acceleration and the actuator displacement. Because of their inherent capability to learn complex nonlinear relationships, neural networks are also effective in nonlinear control problems. In summary, neuro-controllers are effective in structural control and have many advantages over mathematically formulated control algorithms. The performance of neuro-controllers will soon be experimentally verified by the authors in a planned experiment on the three-storey model with the same control system as was used in this study.

The SIMULINK programs which contain the connection weights for the emulator neural network and the three neuro-controllers are available on request from Prof. Jamshid Ghaboussi via e-mail address jghabous @uiuc.edu.

ACKNOWLEDGEMENT

The research reported in this paper was funded by National Science Foundation Grant CMS-95-003209. This support is gratefully acknowledged.

REFERENCES

1. K. Bani-Hani and J. Ghaboussi, 'Nonlinear structural control using neural networks', accepted for publication, *J. Engng Mech. Div. ASCE* **124**(3), 319–328 (1998).
2. J. Ghaboussi, 'Some applications of neural networks in structural engineering', *Proc. Structures Cong. '94*, ASCE, Atlanta, GA, 1994.
3. J. Ghaboussi and K. Bani-Hani, 'Neural network based nonlinear structural control methods', *Proc. 2nd Int. Workshop on Structural Control*, IASC, Hong Kong, 18–21 December, 1996.
4. J. Ghaboussi, and A. Joghataie, 'Active control of structures using neural networks', *J. Engng Mech. Div. ASCE* **121**(4), 555–567 (1995).
5. A. Joghataie and J. Ghaboussi, 'Neural networks and fuzzy logic in structural control', *Proc. 1st World Conf. on Structural Control*, Los Angeles, CA, 1994.
6. A. Joghataie and J. Ghaboussi, 'A comparative study of learning methods and mathematical algorithms in structural control', *Proc. 11th World Conf. on Earthquake Engineering*, Acapulco, Mexico, 1996.
7. K. Nikzad, J. Ghaboussi and S. Paul, 'A study of actuator dynamics and delay compensation using neuro-controllers', *J. Engng Mech. Div. ASCE* **122**, 966–975 (1996).
8. K. Nikzad, 'A study of neural and conventional control paradigms in active digital structural control, Ph.D dissertation, Dept. of Civil Engineering, University of Illinois, Urban-Champaign, 1992.
9. G. W. Housner, S. F. Masri and A. G. Chassiakos (eds) *Proc. 1st World Conf. on Structural Control*, 1 WCSC, 3–5 Aug. 1994, Los Angeles, CA 1994.
10. H. M. Chen, K. H. Tsai, G. Z. Qi and J. C. S. Yang, 'Neural network for structural control', *J. Comput. Civil Engng. ASCE* **9**(2), 168–176 (1995).
11. The Math Works, Inc., *SIMULINK*, Natick, MA, 1994.
12. S. J. Dyke, B. F. Spencer, B. Quast and M. K. Sain, 'The role of control-structure interaction in protective system design', *J. Engng Mech. Div. ASCE* **121**(2) 322–338 (1995).
13. J. Ghaboussi, J. H. Garrett and X. Wu, 'Material modeling with neural networks,' *Proceedings of the Int. Conf. on Numerical Methods in Engineering: Theory and Applications*, Swansea, U.K., 1990, pp. 701–717.
14. J. Ghaboussi, J. H. Garrett and X. Wu, 'Knowledge-based modeling behavior with neural networks', *J. Engng Mech. Div. ASCE* **117**(1), 132–153 (1991).
15. X. Wu, 'Neural network based material modelling, *Ph.D dissertation*, Dept. of Civil Engineering, University of Illinois, Urban-Champaign, 1991.
16. A. Joghataie, J. Ghaboussi and X. Wu, 'Learning and architecture determination through automatic node generation', *Proc. Int. Conf. on Artificial Neural Networks in Engineering*, St. Louis, 1995.
17. S. E. Fahlman, 'Faster learning variations on error backpropagation: an emperical study', *Carnegie-Mellon Summer Workshop on Neural Networks*, 1988.
18. T. Ash, 'Dynamics node creation in backpropagation networks', *Proc. Int. Joint Conf. on Neural Network* (IJCNN), Washington D.C., 1989, June, Vol. II, pp. 623–629.
19. L. L. Chung, A. M. Reinhorn and T. T. Soong, 'Experiments on active control of seismic structures', *J. Engng Mech. Div. ASCE* **114**, 241–256 (1988).

20. The Math Works, Inc., *MATLAB*, Natick, MA, 1994.
21. E. L. Wilson, I. Farhoomand and K. J. Bathe, 'Nonlinear dynamic analysis of complex structures', *Earthquake Engng. Struct. Dyn.* **1**(2), (1972).
22. B. F. Spencer, J. Suhardjo and M. K. Sain, 'Frequency domain optimal control strategies for aseismic protection', *J. Engng Mech. Div. ASCE* **120**, 135–158 (1994).
23. T. T. Soong, *Active Structural Control, Theory and Practice*, Longman Scientific and Technical, 1990.
24. T. T. Soong, A. M. Reinhorn and Y. P. Wang, 'Full scale implementation of active control, design and simulation', *J. Struct. Engng Div. ASCE* **117**, 11, 3516–3536 (1991).

